



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/719,673	11/21/2003	Khosro Khakzadi	03-1862/L13.12-0251	1314

7590 06/23/2009
Leo Peters
LSI Logic Corporation
MS D-106
1621 Barber Lane
Milpitas, CA 95035

EXAMINER

WIENER, ERIC A

ART UNIT	PAPER NUMBER
----------	--------------

2179

MAIL DATE	DELIVERY MODE
-----------	---------------

06/23/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/719,673
Filing Date: November 21, 2003
Appellant(s): KHAKZADI ET AL.

R. Michael Reed
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/11/2009 appealing from the Office action mailed 7/9/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest in contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

An Amendment after Final was filed on 8/12/2009 and approved for entry on 8/27/2008. However, the Amendment did not place the application in condition for allowance.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

Art Unit: 2179

(8) Evidence Relied Upon

US 5,974,253	Nahaboo et al.	10/26/1999
US 5,493,508	Dangelo et al.	2/10/1996

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Art Unit: 2179

3. Claims 5 – 6, 14 – 16, and 21 – 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nahaboo et al. (US 5,974,253) and Dangelo et al. (US 5,493,508).

As per independent claim 1, Nahaboo discloses *a command processor stored on a computer readable memory for use with a computer system (column 6, lines 16 – 18) comprising a graphical user interface program for providing a graphical interface to the computer system (Fig. 2) and a command interpreter (column 6, line 50), which:*

- *loads one or more configuration commands into the command processor from at least one of:*
 - *a user specified command configuration script comprising the one or more configuration commands (column 6, lines 50 – 59 and column 13, lines 32 – 35), wherein command configuration scripts are part of interface description files, which are user specified because a user may specify an interface description file to load, or from*
 - *a command line interface in which the one or more configuration commands are entered by a user (column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig. 4B), wherein the command line area to enter configuration commands is area 315 of Fig. 4B, and*
- *interprets the configuration commands and modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands (column 6, lines 50 – 64), wherein the fact that the interpreter interprets **all** events that arise and that the system reacts in a purely **dynamic** manner means that the interpreter will interpret both saved user*

Art Unit: 2179

specified configuration scripts and configuration commands entered by a user from a command line in order to dynamically modify the interface, *including*:

- *building graphical objects within the graphical interface according to the interpreted configuration commands* (column 7, line 55 – column 8, line 10),
- *assigning functionality to the built graphical objects according to the interpreted configuration commands* (column 9, lines 39 – 42), and
- *displaying the graphical objects within the graphical interface according to the interpreted configuration commands* (column 10, line 45 - column 11, line 2, column 15, lines 20 – 21, and Fig. 5), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface.

However, Nahaboo does not explicitly disclose that the graphical objects are selectable by a user to access the assigned functionality to produce an integrated circuit design.

Nevertheless, in an analogous art, Dangelo discloses *graphical objects selectable by a user to access assigned functionality to produce an integrated circuit design* (column 9, lines 3 – 12), wherein each “component” corresponds to a graphical object with assigned functionality, that is used to produce an integrated circuit design.

As disclosed by Nahaboo's invention is for an extremely flexible interface development tool that can be used *regardless* of the application (column 1, lines 29 – 31). Therefore, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65) and because the design tools of Dangelo's interface may be specified for particular users (Dangelo, column 16, lines 16 – 42), Dangelo would thus look to Nahaboo regarding features of modifying a graphical user interface to produce a design. Furthermore, Nahaboo would look to Dangelo regarding adapting a modifiable graphical user interface to such applications as integrated circuit design, because an integrated circuit design application is one of many possible uses for Dangelo's invention. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo.

As per independent claim 8, Nahaboo discloses *a method of providing a fully customizable graphical user interface* (column 6, lines 11 – 23) *comprising:*

- *upon execution of a command processor by a user, loading a top level Tool Command Language (TCL) command into a namespace* (column 3, lines 52 – 54 and column 6, lines 50 – 56), *the command processor including a graphical user interface (GUI) without graphical objects* (Fig. 4B);
- *loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user* (column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig.

Art Unit: 2179

4B), wherein the command line area to enter configuration commands is area 315 of Fig. 4B, and

- *building graphical objects* (column 7, line 55 – column 8, line 10) *according to the TCL configuration commands* (column 6, lines 53 – 56),
- *assigning functionality to the built graphical objects* (column 9, lines 39 – 42) *according to the TCL configuration commands* (column 6, lines 53 – 56), and
- *displaying the graphical objects within the GUI according to the TCL configuration commands* (column 3, lines 34 – 36), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface.

The examiner has interpreted the fact that Nahaboo discloses using an interpreted language (Abstract, lines 1 – 2) sufficiently discloses the use of the interpreted language Tool Command Language.

However, Nahaboo does not explicitly disclose that the graphical objects are selectable by a user to produce an integrated circuit design.

Nevertheless, in an analogous art, Dangelo discloses *graphical objects selectable by a user to produce an integrated circuit design* (column 9, lines 3 – 12), wherein each “component” corresponds to a graphical object with assigned functionality that is used to produce an integrated circuit design.

It would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*.

As per independent claim 19, Nahaboo discloses *a method of providing a graphical user interface, comprising:*

- *loading a top level Tool Command Language (TCL) command into a namespace upon execution of a command processor (column 3, lines 52 – 54 and column 6, lines 50 – 56)*
- *providing a command interpreter for interpreting one or more configuration commands from a user (column 6, lines 50 – 52)*
- *loading a configuration command of the one or more configuration commands into the command processor from at least one of:*
 - *a user specified command configuration script comprising the configuration command (column 6, lines 50 – 59 and column 13, lines 32 – 35), wherein command configuration scripts are part of interface description files, which are user specified because a user may specify an interface description file to load, or from*
 - *a command line in which the configuration command is entered by the user (column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig. 4B), wherein the command line area to enter configuration commands is area 315 of Fig. 4B, and*

Art Unit: 2179

- *assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user (column 6, lines 53 – 59)*

- *wherein all objects within the graphical user interface are user defined through the one or more configuration commands (column 6, lines 53 – 59), and wherein the fact that the user can enter the ‘editing’ mode without entering the ‘execution’ mode means that said user can define all objects of the graphical user interface before execution, thus defining all objects of a graphical user interface having no hard coded objects.*

The examiner has interpreted the fact that Nahaboo discloses using an interpreted language (Abstract, lines 1 – 2) sufficiently discloses the use of the interpreted language Tool Command Language.

However, Nahaboo does not explicitly disclose that the at least one graphical user interface (GUI) component is selectable by a user to access an associated function to generate an integrated circuit design.

Nevertheless, in an analogous art, Dangelo discloses *at least one graphical user interface (GUI) component is selectable by a user to access an associated function to generate an integrated circuit design* (column 9, lines 3 – 12), wherein each “component” corresponds to a GUI component with assigned functionality that is used to produce an integrated circuit design.

It would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*.

As per independent claim 24, Dangelo discloses *an integrated circuit software design suite* (column 5, lines 21 – 26) *comprising*:

- *a command processor having a graphical user interface* (column 8, lines 47 – 67) *and a command interpreter* (column 5, lines 25 – 28) *for interpreting user commands*

wherein the fact that the system processes user selections for graphically interfacing with elements sufficiently corresponds to having a processor for processing and interpreting commands

- *one or more design tools corresponding to processes within an integrated circuit design process* (column 8, lines 47 – 50)

- *wherein the one or more design tools operate under control of the command processor and within the graphical user interface* (column 3, lines 46 – 61 and column 8, lines 47 – 67), *further wherein the configuration commands build graphical objects for the graphical user interface and assign functionality to the built graphical objects* (column 3, lines 34 – 61; column 8, lines 47 – 67; and column 9, line 54 – column 10, line 10), wherein column 3, lines 46 – 61 discloses that the description supports a structural description of graphical components that corresponds to a behavioral description of program statements.

Dangelo does not explicitly disclose that the graphical user interface is specified entirely by a user via a user input including one or more configuration commands provided to the command processor at runtime.

However, in an analogous art, Nahaboo discloses *a graphical user interface specified entirely by a user via user input through one or more configuration commands loaded into the command processor at run time of the command processor and interpreted by the*

command interpreter (column 6, lines 56 – 64), wherein the configuration commands build graphical objects for the graphical user interface (column 7, line 55 – column 8, line 10) and assign functionality to the built graphical objects (column 9, lines 39 – 42).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Dangelo with the teachings of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*. Furthermore, since the user is able to specify parts of the interface at runtime, it would be obvious that said user is also able to specify the entire interface at runtime.

As per claim 5, and taking into account the rejection of claim 1, Dangelo further discloses *a suite of integrated circuit tools, each design tool of the suite having a functionality corresponding to one or more steps in a design flow process of an integrated circuit (column 1, lines 7 – 9, 38 – 41), wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate this teaching of Dangelo into the invention of Nahaboo for the same rationale as disclosed in the rejection of claim 1, supra.*

As per claim 6, and taking into account the rejection of claim 5, Nahaboo further discloses that *the command processor loads each design tool into the graphical user interface based on the user configuration commands (column 6, lines 50 – 52).*

As per claim 7, and taking into account the rejection of claim 1, Nahaboo further discloses *a graphics engine tool for drawing contents of a database into the graphical user interface based on the user configuration commands (column 3, lines 52 – 61).*

As per claim 9, and taking into account the rejection of claim 8, Nahaboo further discloses *performing functions based on user interactions with the graphical objects according to their assigned functionality* (column 10, lines 45 – 54).

As per claim 10, and taking into account the rejection of claim 8, Nahaboo further discloses that *the graphical objects are selected from a group consisting of windows, window panes* (column 4, lines 39 – 41), *buttons, and menus* (column 6, lines 4 – 5).

As per claim 11, and taking into account the rejection of claim 8, Nahaboo further discloses *creating the TCL command configuration script and assigning the TCL command configuration script to one of the graphical objects* (column 11, lines 55 – 58). Nahaboo does not explicitly disclose the TCL script corresponds to a circuit design function. However, Nahaboo does disclose that “the purpose of this invention is to define an extremely flexible interface development tool that can be used regardless of the application” (column 1, lines 29 – 31). Therefore, Nahaboo’s script could correspond to a circuit design function if the user defines it in such a way.

As per claim 12, and taking into account the rejection of claim 11, Nahaboo further discloses that *one of the graphical objects is a button* (column 6, line 4).

As per claim 13, and taking into account the rejection of claim 11, Nahaboo further discloses that *one of the graphical objects is an item within a pull-down menu* (column 6, line 5).

As per claim 14, and taking into account the rejection of claim 8, Dangelo further discloses *changing a look and feel of the graphical user interface during a circuit design process* (column 2, lines 62 – 65), wherein it would have been obvious to one of ordinary

Art Unit: 2179

skill in the art at the time of invention to incorporate this teaching of Dangelo into the invention of Nahaboo for the same rationale as disclosed in the rejection of claim 1, *supra*.

As per claim 15, and taking into account the rejection of claim 14, Nahaboo further discloses *creating new graphical objects, previously undefined by the command processor* (column 7, line 55 – column 8, line 10) *using the TCL configuration commands* (column 6, lines 53 – 56) *and assigning functionality to the new graphical objects* (column 9, lines 39 – 42).

As per claim 16, and taking into account the rejection of claim 14, Nahaboo further discloses:

- *loading a new top level TCL command into the namespace which corresponds to one or more new TCL configuration commands* (column 3, lines 52 – 54, column 6, lines 50 – 56, and column 9, lines 53 – 60)
- *building graphical objects* (column 7, line 55 – column 8, line 10 and column 10, lines 36 – 44) *according to new TCL configuration commands* (column 6, lines 53 – 56)
- *assigning functionality to the built graphical objects* (column 9, lines 39 – 42 and column 10, lines 36 – 44) *according to the new TCL configuration commands* (column 6, lines 53 – 56)
- *displaying the user-interactive window containing the graphical objects according to the new TCL configuration commands* (column 3, lines 34 – 36), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration

Art Unit: 2179

commands as made possible through Nahaboo's disclosed invention for developing a user interface

The examiner has interpreted the fact that the interface is able to load the file of commands to reconstruct the interface in a purely dynamic manner sufficiently corresponds to being able to change a look and feel according to new commands and objects.

As per claim 20, and taking into account the rejection of claim 19, Nahaboo further discloses *changing the graphical user interface based on changing configuration commands from the user* (column 6, lines 62 – 63) *and displaying a changed graphical user interface during operation based on the changing configuration commands* (column 6, lines 56 – 59).

As per claim 21, and taking into account the rejection of claim 19, Dangelo further discloses *interfacing with a suite of integrated circuit design tools for producing an integrated circuit layout and associated netlist* (column 2, line 55 – column 3, line 13), wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*. Furthermore, since the user has control over defining some aspects of an interface, it would be obvious that the user also has the ability to completely define all aspects of an interface for circuit design.

As per claim 22, and taking into account the rejection of claim 21, Nahaboo further discloses *loading a design tool from the suite of design tools into the graphical user interface based on a user command* (column 8, lines 7 – 51).

As per claim 23, and taking into account the rejection of claim 22, Nahaboo further discloses that *the one or more user configuration commands are assigned to one or more graphical objects* (column 9, lines 39 – 42).

As per claim 25, and taking into account the rejection of claim 24, Dangelo further discloses that *the command processor interprets the user input to generate at least one graphical object within the graphical user interface associated with at least one design tool of the one or more design tools* (column 3, lines 34 – 61; column 8, lines 47 – 67; and column 9, line 54 – column 10, line 10), wherein column 3, lines 46 – 61 discloses that the description supports a structural description of graphical components that corresponds to a behavioral description of program statements, further wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*.

As per claim 26, and taking into account the rejection of claim 25, Dangelo further discloses that *the graphical object is selectable by the user to load the at least one design tool into the graphical user interface, the graphical user interface accessible by the user to produce an integrated circuit design* (column 14, lines 22 – 39 and 51 – 59), wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, *supra*.

As per claim 27, and taking into account the rejection of claim 5, Dangelo further discloses that *at least one of the graphical objects is associated with at least one integrated*

Art Unit: 2179

design tool of the suite of integrated circuit design tools (column 13, lines 54 – 61), wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, supra.

As per claim 28, and taking into account the rejection of claim 27, Dangelo further discloses that *the at least one integrated design tool is executable by the command interpreter to design and test an integrated circuit layout, and wherein the at least one of the graphical objects is selectable by the user to access the at least one integrated design tool (column 14, lines 22 – 39 and 51 – 59), wherein it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo for the same reasons as disclosed in the rejection of claim 1, supra.*

10) Response to Arguments

Appellant asserts the following:

1. The Appellant has argued that *"with respect to claim 8, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest "upon execution of a command processor by a user, loading a top level Tool Command Language (TCL) command into a namespace, the command processor including a graphical user interface (GUI) without graphical objects," "loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user," "building graphical objects according to the TCL configuration commands," "assigning functionality to the built graphical objects according to the TCL*

configuration commands", and "displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated circuit design", as recited in claim 8."

In response to this argument, the Examiner respectfully disagrees. Please refer to column 3, lines 52 – 54 and column 6, lines 50 – 56 of Nahaboo, which sufficiently teach building and assigning functionality to objects, wherein the use of an object toolbox, such as the disclosed X/MOTIF graphical object toolbox, has been interpreted as sufficiently corresponding to the use of a Tool Command Language, wherein both are languages that may be interchangeably used for creating, modifying, and displaying an interface, and in either case will achieve the same desired result.

Furthermore, Fig. 4B depicts “including a graphical user interface (GUI) without graphical objects.”

In addition, please refer to column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig. 4B of Nahaboo as disclosing loading of one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user, wherein the command line area to enter configuration commands is area 315 of Fig. 4B.

In addition please refer to column 7, line 55 – column 8, line 10 and column 6, lines 53 – 56 of Nahaboo as disclosing building graphical objects according to the TCL configuration commands, wherein assigning functionality to the built graphical objects is further disclosed in column 9, lines 39 – 42.

Furthermore, please also refer to column 3, lines 34 – 36; column 10, line 45 – column 11, line 2; column 15, lines 20 – 21; and Fig. 5, wherein an example of displaying a user-interactive window containing the graphical objects according to the interpreted configuration commands is disclosed, further wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface. Furthermore, Fig. 4B allows for configuration commands to be loaded by the user through a command line during execution. In addition, Nahaboo's invention is for an extremely flexible interface development tool that can be used regardless of the application, and thus is able to be used to produce an integrated circuit design if the user so wishes (column 1, lines 29 – 31).

2. The Appellant has argued that *“though the design tool of Nahaboo allows a developer to alter the user interface during design, there is no suggestion or teaching that the user interface can be modified during operation by an end user (who is an integrated circuit designer and not the software developer).”*

The Examiner respectfully disagrees. In response to Appellant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, please refer to Nahaboo, column 6, lines 50-64 and column 7, line 55 - column 8, line 10, wherein it has been interpreted that the fact that the interpreter interprets all events that arise and that the system reacts in a purely dynamic manner means that the interpreted will interpret both saved user specified configuration scripts and configuration commands entered by a user from a command line in order to dynamically, and thus at run time, modify the interface. Furthermore, it has been interpreted that the editor serves to modify at run time, because a user edits the objects directly through the interface, and thus at a run time of the interface.

Furthermore, Nahaboo's invention is for an extremely flexible interface development tool that can be used regardless of the application, and thus is able to be used to produce an integrated circuit design if the user so wishes (column 1, lines 29 – 31).

3. The Appellant has argued that *“Nahaboo discloses an interface development tool to design and refine interfaces that can be used with applications, but does not suggest that the user of the applications allow for design and refinement of the interface "at run time of the graphical user interface," as recited in claim 1.”*

In response to this argument, the Examiner respectfully disagrees. Please refer to Nahaboo, column 6, lines 50-64 and column 7, line 55 - column 8, line 10, wherein it has been interpreted that the fact that the interpreter interprets all events that arise and that the system reacts in a purely dynamic manner means that the interpreted will interpret both saved user specified configuration scripts and configuration commands entered by a user

from a command line in order to dynamically, and thus at run time, modify the interface. Furthermore, it has been interpreted that the editor serves to modify at run time, because a user edits the objects directly through the interface, and thus at a run time of the interface.

4. The Appellant has argued that *“the interface development tool of Nahaboo is not also used to produce an integrated circuit design.”*

In response to this argument, the Examiner respectfully disagrees. Please refer to the response to Appellant’s arguments pertaining to section 2., *supra*, which discloses these features.

5. The Appellant has argued that *“Nahaboo provides no suggestion or motivation for combining the interface development tool with the actual application that is executed by the user (and not the designer). Further, Nahaboo provides no suggest or teaching that the “loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user,” “building graphical objects according to the TCL configuration commands,” “assigning functionality to the built graphical objects according to the TCL configuration commands”, and “displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated circuit design”, as recited in claim 8.”*

In response to this argument, the Examiner respectfully disagrees. Please refer to the response to Appellant's arguments pertaining to section 1., *supra*, which discloses these features.

6. The Appellant has argued that *"Dangelo fails to disclose or suggest 'building graphical objects within the graphical interface according to the interpreted configuration commands,' as recited in claim 8. Further, Dangelo does not disclose or suggest 'upon execution of a command processor, loading a top level Tool Command Language (TCL) command into a namespace, the command processor including a graphical user interface (GUI) without graphical objects,' as recited in claim 8."*

The Examiner respectfully disagrees. In response to Appellant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, Please refer to the response to Appellant's arguments pertaining to section 1., *supra*, wherein it is disclosed that Nahaboo discloses and makes obvious such features.

7. The Appellant has argued that *"Dangelo does not disclose a 'command processor including a graphical user interface (GUI) without graphical objects,' and does not disclose 'loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the*

user," "building graphical objects according to the TCL configuration commands," "assigning functionality to the built graphical objects according to the TCL configuration commands", and "displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated circuit design", as recited in claim 8."

The Examiner respectfully disagrees. In response to Appellant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, please refer to the response to Appellant's arguments pertaining to section 1., *supra*, wherein it is disclosed that Nahaboo discloses and makes obvious such features.

8. The Appellant has argued that an *"impermissible hindsight reconstruction is present here, where the Office ignores the explicit teachings of Nahaboo relating to separate applications including 1) an interface development tool to construct user interfaces and 2) an application with an embedded WOOL interpreter program to utilize the created interfaces to produce an interface for use with the application. The design tool is to be used by a software designer, and not an end user. Further, the WOOL interpreter program is not indicated to allow a user to specify or control the interface, but rather Nahaboo discloses that the interface development tool is used to modify the interface. The*

interface development tool is indicated to be usable with a variety of applications, specifically because the WOOL interpreter program is embedded with each of the applications to load the interface file. Notwithstanding the suggestion of the Office that the design application of Dangelo could be combined with the interface development tool, the teachings of Nahaboo teach away from such a combination. The only motivation for modifying either Nahaboo or Dangelo to provide the elements disclosed in the claims is provided by the present application.”

In response to this argument, the Examiner respectfully disagrees. It has been determined that, because Nahaboo's invention is for an extremely flexible interface development tool that can be used regardless of the application (column 1, lines 29 – 31), and further because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65), and further because the design tools of Dangelo's interface may be specified for particular users (Dangelo, column 16, lines 16 – 42), Dangelo would thus look to Nahaboo regarding features of modifying a graphical user interface to produce a design. In addition, Nahaboo would look to Dangelo regarding adapting a modifiable graphical user interface to such applications as integrated circuit design, because an integrated circuit design application is one of many possible uses for Dangelo's invention. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo.

9. The Appellant has argued that “*Dangelo does not disclose or suggest a user specified GUI.*”

The Examiner respectfully disagrees. In response to Appellant’s arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Therefore, please also refer to Nahaboo, column 6, lines 56 – 64, as disclosing a graphical user interface specified entirely by a user via user input through one or more configuration commands loaded into the command processor at run time of the command processor and interpreted by the command interpreter, wherein the configuration commands build graphical objects for the graphical user interface (column 7, line 55 – column 8, line 10) and assign functionality to the built graphical objects (column 9, lines 39 – 42).

10. The Appellant has argued that “*while Nahaboo recognizes a problem with poor interface design, Nahaboo does not disclose or suggest the interface design tool be combined with the application.*”

The Examiner respectfully disagrees. In response to Appellant’s arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See

In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); In re Merck & Co., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Therefore, please also refer to Dangelo, column 3, lines 46 – 61 and column 8, lines 47 – 67, wherein the one or more design tools operate under control of the command processor and within the graphical user interface, further wherein the configuration commands build graphical objects for the graphical user interface and assign functionality to the built graphical objects (column 3, lines 34 – 61; column 8, lines 47 – 67; and column 9, line 54 – column 10, line 10), thus serving to be combined with an application.

11. The Appellant has argued that *“neither Nahaboo nor Dangelo provide any suggestion or motivation to incorporate an interface design tool with an application that generates integrated circuit designs. Further, Nahaboo provide no suggestion or motivation for incorporating its interface design tool into the application, but rather teaches away from such incorporation by teaching embedding of a WOOL interpreter program instead. Accordingly, there exists no motivation for making the asserted combination. Therefore, the asserted combination of Nahaboo and Dangelo constitutes an impermissible hindsight reconstruction, and the rejection of the pending claims over the asserted combination of Nahaboo and Dangelo should be withdrawn.”*

In response to this argument, the Examiner respectfully disagrees. Please refer to the response to Appellant’s arguments pertaining to section 8., *supra*, wherein it is disclosed that it would have been obvious to one of ordinary skill in the art at the time of

Art Unit: 2179

invention to incorporate the teachings of Dangelo into the invention of Nahaboo without any such hindsight reasoning.

12. The Appellant has argued that *"the asserted combination of Nahaboo and Dangelo fails to disclose or suggest "the graphical user interface specified entirely by a user via a user input including one or more configuration commands provided to the command processor at run time of the command processor and interpreted by the command interpreter, wherein the configuration commands build graphical objects within the graphical user interface and assign functionality to the built graphical objects" and "one or more design tools corresponding to processes within an integrated circuit design process", "wherein the one or more design tools operate under control of the command processor and within the graphical user interface", as recited in claim 24."*

In response to this argument, the Examiner respectfully disagrees. Please refer to Nahaboo, column 6, lines 50-64 and column 7, line 55 - column 8, line 10, wherein it has been interpreted that the fact that the interpreter interprets all events that arise and that the system reacts in a purely dynamic manner means that the interpreted will interpret both saved user specified configuration scripts and configuration commands entered by a user from a command line in order to dynamically, and thus at run time, modify the interface. Furthermore, it has been interpreted that the editor serves to modify at run time, because a user edits the objects directly through the interface, and thus at a run time of the interface.

Furthermore, Nahaboo's invention is for an extremely flexible interface development tool that can be used regardless of the application, and thus is able to be used to produce an integrated circuit design if the user so wishes (column 1, lines 29 – 31).

13. The Appellant has argued that *"Dangelo fails to disclose or suggest 'the graphical user interface specified entirely by a user via a user input including one or more configuration commands...', as recited in claim 19."*

The Examiner respectfully disagrees. In response to Appellant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, please refer to Nahaboo, column 6, lines 53 – 59, wherein all objects within the graphical user interface are user defined through the one or more configuration commands, because the fact that the user can enter the 'editing' mode without entering the 'execution' mode has been interpreted to mean that said user can define all objects of the graphical user interface before execution, thus defining all objects of a graphical user interface having no hard coded objects.

14. The Appellant has argued that *"the asserted combination of Nahaboo and Dangelo fails to disclose or suggest 'the graphical user interface specified entirely by a user via a user input including one or more configuration commands provided to the command*

Art Unit: 2179

processor at run time of the command processor and interpreted by the command interpreter, wherein the configuration commands build graphical objects within the graphical user interface and assign functionality to the built graphical objects" and "wherein the one or more design tools operate under control of the command processor and within the graphical user interface", as recited in claim 24."

In response to this argument, the Examiner respectfully disagrees. Nahaboo discloses a graphical user interface specified entirely by a user via user input through one or more configuration commands loaded into the command processor at run time of the command processor and interpreted by the command interpreter (column 6, lines 56 – 64), wherein the configuration commands build graphical objects for the graphical user interface (column 7, line 55 – column 8, line 10) and assign functionality to the built graphical objects (column 9, lines 39 – 42).

15. The Appellant has argued that *"the asserted combination of Nahaboo and Dangelo fails to disclose or suggest "a command interpreter" that "modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands" including "building graphical objects within the graphical interface according to the interpreted configuration commands," "assigning functionality to the built graphical objects according to the interpreted configuration commands," and "displaying the graphical objects within the graphical interface according to the interpreted configuration commands," "wherein the graphical objects are selectable by a user to access the assigned functionality to produce an integrated circuit design", as recited in claim 1."*

In response to this argument, the Examiner respectfully disagrees. In column 6, lines 50 – 64, Nahaboo discloses a command interpreter, which interprets the configuration commands and modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands, wherein the fact that the interpreter interprets all events that arise and that the system reacts in a purely dynamic manner means that the interpreter will interpret both saved user specified configuration scripts and configuration commands entered by a user from a command line in order to dynamically modify the interface.

In addition Nahaboo further discloses building graphical objects within the graphical interface according to the interpreted configuration commands (column 7, line 55 – column 8, line 10), assigning functionality to the built graphical objects according to the interpreted configuration commands (column 9, lines 39 – 42), and displaying the graphical objects within the graphical interface according to the interpreted configuration commands (column 10, line 45 - column 11, line 2, column 15, lines 20 – 21, and Fig. 5), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface.

Furthermore, Nahaboo's invention is disclosed as an extremely flexible interface development tool that can be used regardless of the application (column 1, lines 29 – 31). In addition, Dangelo discloses graphical objects selectable by a user to access assigned functionality to produce an integrated circuit design (column 9, lines 3 – 12), wherein

each “component” corresponds to a graphical object with assigned functionality that is used to produce an integrated circuit design.

Therefore, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65) and because the design tools of Dangelo’s interface may be specified for particular users (Dangelo, column 16, lines 16 – 42), Dangelo would thus look to Nahaboo regarding features of modifying a graphical user interface to produce a design. Furthermore, Nahaboo would look to Dangelo regarding adapting a modifiable graphical user interface to such applications as integrated circuit design, because an integrated circuit design application is one of many possible uses for Dangelo’s invention. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo.

16. The Appellant has argued that “*Nahaboo discloses an interface development tool to design and refine interfaces that can be used with applications, but does not suggest that the user of the applications allow for design and refinement of the interface "at run time of the graphical user interface," as recited in claim 1,*” and that “*Nahaboo does not disclose or suggest that the WOOL language interpreter "modifies the graphical user interface at run time of the graphical user interface", as recited in claim 1. Accordingly, Nahaboo does not disclose or suggest "a command interpreter" that "modifies the graphical user interface at run time of the graphical user interface", as recited in claim 1. While Dangelo discloses that the schematic editors allow for selection of previously*

created and stored schematics, Dangelo does not disclose or suggest "a command interpreter" that "modifies the graphical user interface at run time of the graphical user interface", as recited in claim 1."

The Examiner respectfully disagrees. In response to Appellant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, please refer to Nahaboo, column 6, lines 50-64 and column 7, line 55 - column 8, line 10, wherein it has been interpreted that the fact that the interpreter interprets all events that arise and that the system reacts in a purely dynamic manner means that the interpreted will interpret both saved user specified configuration scripts and configuration commands entered by a user from a command line in order to dynamically, and thus at run time, modify the interface. Furthermore, it has been interpreted that the editor serves to modify at run time, because a user edits the objects directly through the interface, and thus at a run time of the interface.

Also, please refer to column 7, line 55 – column 8, line 10; column 9, lines 39 – 42; and column 10, lines 36 – 44 of Nahaboo, wherein the creation of objects has been interpreted as sufficiently corresponding to building objects, and further wherein the commands pertaining to said creation, which assign functionality, may be interpreted configuration commands, such as those entered by a user.

Furthermore, please refer to column 3, lines 52 – 54 and column 6, lines 50 – 56 of Nahaboo, which sufficiently teach building and assigning functionality to objects, wherein the use of an object toolbox, such as the disclosed X/MOTIF graphical object toolbox, has been interpreted as sufficiently corresponding to the use of a Tool Command Language, wherein both are languages that may be interchangeably used for creating, modifying, and displaying an interface, and in either case will achieve the same desired result.

17. The Appellant has argued that *"the asserted combination of Nahaboo and Dangelo fails to disclose or suggest "loading a configuration command of the one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the configuration command or from a command line in which the configuration command is entered by the user" and "assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design", as recited in claim 19."*

In response to this argument, the Examiner respectfully disagrees. Please refer to column 6, lines 50 – 59 and column 13, lines 32 – 35 of Nahaboo as disclosing loading a configuration command of the one or more configuration commands into the command processor from a user specified command configuration script comprising the configuration command wherein command configuration scripts are part of interface

description files, which are user specified because a user may specify an interface description file to load.

In addition, please refer to column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig. 4B of Nahaboo as disclosing loading a configuration command of the one or more configuration commands into the command processor from a command line in which the configuration command is entered by the user, wherein the command line area to enter configuration commands is area 315 of Fig. 4B.

Furthermore, please refer to column 6, lines 53 – 59 of Nahaboo as disclosing assembling a graphical user interface having no hard coded objects that includes at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, wherein all objects within the graphical user interface are user defined through the one or more configuration commands, and wherein the fact that the user can enter the ‘editing’ mode without entering the ‘execution’ mode means that said user can define all objects of the graphical user interface before execution, thus defining all objects of a graphical user interface having no hard coded objects.

Even though Nahaboo does not explicitly disclose that the at least one graphical user interface (GUI) component is selectable by a user to access an associated function to generate an integrated circuit design, in an analogous art, column 9, lines 3 – 12 of Dangelo does disclose that at least one graphical user interface (GUI) component is selectable by a user to access an associated function to generate an integrated circuit

design, wherein each “component” corresponds to a GUI component with assigned functionality that is used to produce an integrated circuit design.

Therefore, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65) and because the design tools of Dangelo’s interface may be specified for particular users (Dangelo, column 16, lines 16 – 42), Dangelo would thus look to Nahaboo regarding features of modifying a graphical user interface to produce a design. Furthermore, Nahaboo would look to Dangelo regarding adapting a modifiable graphical user interface to such applications as integrated circuit design, because an integrated circuit design application is one of many possible uses for Dangelo’s invention. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teachings of Dangelo into the invention of Nahaboo.

18. The Appellant has argued that *“while Dangelo discloses that circuit specifications can be reused, Dangelo does not disclose or suggest “assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design”, as recited by claim 19.”*

The Examiner respectfully disagrees. In response to Appellant’s arguments against the references individually, one cannot show nonobviousness by attacking

Art Unit: 2179

references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In addition, please refer to the response to Appellant's arguments pertaining to section 17., *supra*, wherein it is disclosed that the combination of Nahaboo and Dangelo discloses and makes obvious such features.

(11) Related Proceedings Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Eric Wiener/

Examiner, Art Unit 2179

Conferees:

/Ba Huynh/

Primary Examiner, Art Unit 2179

/Weilun Lo/

Supervisory Patent Examiner, Art Unit 2179